USEing Transfer Learning in Retrieval of Statistical Data

July 24, 2019

Anton Firsov, Vladimir Bugay, Anton Karpenko – Knoema Corporation
INTRODUCTION

- Knoema is a global data aggregator and a search engine for data
- Our search operates with 3.2B time series which are mostly numbers with a limited textual metadata available
- More than 500K analysts and researchers look for data, facts and insights at https://knoema.com every month

HOW MANY PEOPLE LIVE IN PARIS?

WHAT IS CHILD MORTALITY IN UGANDA

HOW MUCH MONEY IS SPENT ON RESEARCH IN USA
SPECIFICS OF OUR DOMAIN

- Narrow domain => less users => less data for training models
- Very short documents – time series
- Structure in the textual metadata (multiple fields aka dimensions, hierarchies)
- Complex queries for which only a collection of related time series can be an answer

Paris – Population

Uganda – Under-5 mortality rate (per 1,000 live births)

United States – Basic Research Expenditures, Public Research, Million USD PPPs
COMPARISON QUERIES

CHINA VS INDIA POPULATION

INSTANT ANSWER

TIME SERIES
COUNTRIES BY GDP PER CAPITA

Gross domestic product per capita in current prices (US dollars)

Luxembourg is the top country by GDP per capita in the world. As of 2018, GDP per capita in Luxembourg was 114,234 US dollars. The top 5 countries also includes Switzerland, Macau, Norway, and Ireland.

The description is composed by our digital data assistant.

GDP per capita is gross domestic product divided by midyear population. GDP is the sum of gross value added by all resident producers in the economy plus any product taxes and minus any subsidies not included in the value of the products. It is calculated without making deductions for depreciation of fabricated assets or for depletion and degradation of natural resources. GDP is expressed in current U.S. dollars per person. Data are derived by first converting GDP in national currency to U.S. dollars and then dividing by total population.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Luxembourg</td>
<td>114,234</td>
<td>105,713</td>
<td>102,361</td>
<td>101,665</td>
<td>120,450</td>
<td>114,998</td>
<td>108,048</td>
<td>117,340</td>
<td>106,185</td>
<td>81,105</td>
<td>49,173</td>
<td>35,201</td>
</tr>
<tr>
<td>Switzerland</td>
<td>89,940</td>
<td>80,643</td>
<td>80,491</td>
<td>82,510</td>
<td>87,167</td>
<td>85,676</td>
<td>83,959</td>
<td>88,903</td>
<td>74,885</td>
<td>55,115</td>
<td>37,994</td>
<td>38,660</td>
</tr>
<tr>
<td>Macau SAR</td>
<td>82,388</td>
<td>77,415</td>
<td>70,278</td>
<td>70,133</td>
<td>86,931</td>
<td>84,360</td>
<td>73,938</td>
<td>65,868</td>
<td>50,921</td>
<td>24,970</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Norway</td>
<td>81,695</td>
<td>75,514</td>
<td>70,703</td>
<td>74,281</td>
<td>96,838</td>
<td>102,722</td>
<td>101,273</td>
<td>100,307</td>
<td>87,432</td>
<td>66,653</td>
<td>38,063</td>
<td>28,730</td>
</tr>
</tbody>
</table>
PREVIOUS APPROACHES

INVERTED INDEX + ONTOLOGY + PRE/POST PROCESSING

- Requires domain-specific ontology
- Problem with on-site data repositories
- A lot of heuristics and parameters => difficult to maintain

DEEP STRUCTURED SEMANTIC MODEL (DSSM) [Huang et al., 2013]

- Small amount of click through data (~100K)
## TRANSFER LEARNING

<table>
<thead>
<tr>
<th></th>
<th><strong>USE</strong></th>
<th><strong>BERT</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Full name</strong></td>
<td>Universal Sentence Encoder</td>
<td>Bidirectional Encoder Representations from Transformers</td>
</tr>
<tr>
<td><strong>Authors</strong></td>
<td>[Cer et al., 2018]</td>
<td>[Devlin et al., 2018]</td>
</tr>
<tr>
<td><strong>Model variation</strong></td>
<td>transformer-based</td>
<td>base, uncased model for English language</td>
</tr>
<tr>
<td><strong>Underlying architecture</strong></td>
<td>Transformer</td>
<td>Transformer</td>
</tr>
<tr>
<td><strong>Number of attention layers</strong></td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td><strong>Number of parameters</strong></td>
<td>~200M</td>
<td>~110M</td>
</tr>
<tr>
<td><strong>Embedding size</strong></td>
<td>512</td>
<td>768</td>
</tr>
</tbody>
</table>
ARCHITECTURE

P(D|Q)

Cosine similarity

USE query model
Query (Q)

USE document model
Document (D)
MODEL

- $\bar{Q} = USE_Q(Q)$, where $Q$ is the query and $USE_Q$ - USE query model
- $\bar{D} = USE_D(D)$, where $D$ is the document (timeseries), $USE_D(D)$ - USE document model
- $S(Q, D) = \cos(\bar{Q}, \bar{D})$ – similarity between a query $Q$ and document $D$

To effectively calculate probability of document $D$ given query $Q$ we used negative sampling:

- $D = \{D^+, D_1^-, D_2^-, ..., D_k^-\}$, where $D^+$ is the document that was clicked for query $Q$ and $D_i^-$ - random unclicked documents
- $P(D^+|Q) = \frac{\exp(S(Q,D^+))}{\sum_{D_i \in D} \exp(S(Q,D_i))}$
- $loss(Q, D) = -\log(P(D^+|Q))$
IMPLEMENTATION

Click data
BERT/USE

Fine-tuning

Finetuned model

Timeseries

Embeddings calculation

Embeddings

Indexing

Index

Query

Search
TRAINING

- Training set ~13K click-through samples
- CV set ~2K click-through samples
- Adam optimizer with learning rate $1 \times 10^{-5}$
- Batch size 32 (BERT) and 128 (USE)
- 4 negative samples per query
- 600 steps
- Training time <5 min on V100
# EMBEDDING CALCULATIONS

## USE
- 400M timeseries
- Calculated on V100
- ~8K timeseries per second
- Total time: ~14 hours
- Cost: ~32$
- Total size: ~900Gb

## BERT
- 400M timeseries
- Calculated on Google TPUv3
- ~10K timeseries per second
- Total time: ~11 hours
- Cost: ~90$
- Total size: ~1.3Tb
Using FAISS library [Johnson et al., 2017] for an approximate nearest neighbor search

- IVF index with $2^{18}$ centroids and HNSW quantizer
- Centroids are trained on 25M random vectors ($\sim 5h$ on r5.2xlarge)
- Product Quantization with 32 and 16 components for index size reduction
- Total time to build index: $\sim 10h$
- Index size $\sim 17Gb$ for PQ=32 and $\sim 11Gb$ for PQ=16
A/B test: mixed equal number of classic and USE results

<table>
<thead>
<tr>
<th>Source of clicked results</th>
<th>Number of clicked results</th>
</tr>
</thead>
<tbody>
<tr>
<td>USE</td>
<td>2791</td>
</tr>
<tr>
<td>Classic</td>
<td>2352</td>
</tr>
<tr>
<td>Total</td>
<td>5143</td>
</tr>
</tbody>
</table>

18% HIGHER CTR
RESULT ANALYSIS

AUTOMATICALLY DEDUCED SEMANTIC CLOSENESS

- **Query:** us gdp
- **Result:** United States - Gross domestic product, current prices (U.S. dollars)

QUESTIONS IN NATURAL LANGUAGE

- **Query:** how many people live in paris?
- **Result:** Paris - Population

RESULT GENERALIZATION

- **Query:** bmw theft in japan
- **Result:** Japan - Theft of Private Cars - Rate
WHAT’S NEXT

COMPLEX QUERIES PROCESSING

- "china vs india population"
- "countries ranking by gdp"
- "world population density in 2017 on map"

CHATBOT (DIGITAL RESEARCH ASSISTANT)

- Need to keep context of the conversation
- Difficulties with general questions

RETRIEVAL OF STATISTICAL DATA RELEVANT TO THE TEXT (FACTFINDER)

- Multiple vectors per text
- Co-reference, ellipsis, anaphora, endophora resolution

SUPPORT OF MULTIPLE LANGUAGES
Finetuning of pretrained deep neural net models allowed us to:

- Improve the results of our search engine
- Decrease cost of ontology engineering
- Decrease resources cost (memory and CPU)
- Continuously, automatically and cost-effectively improve our search engine further using clickstream data
- Reduce codebase and simplify its maintenance

However, some tasks such as complex query processing are still easier to solve with heuristics and some pre/post processing
THANK YOU FOR ATTENTION!

QUESTIONS?

ANTON FIRSOV

AFIRSOV@KNOEMA.COM